EDN | COLUMNIST

DAVID BRUBAKER,
FUZZY-LOGIC CONTRIBUTING EDITOR

Dismantling a fuzzy rule-based system shows its inner workings

Stop me if you've heard this one. The French Revolution is in full swing. The government has condemned three people—a physician, a lawyer, and an engineer—to the guillotine. As a statement of their innocence, all three elect to lie on their backs, facing upward. The lawyer goes first. As the blade falls, it snags and stops six inches above his neck. The convention is that a guillotine malfunction indicates divine intervention, and the lawyer goes free.

The executioner makes a few adjustments, and it is the physician's turn. Again, the blade stops short, and the physician also goes free. Now somewhat frustrated and perhaps out of his league, the executioner again fiddles with the guillotine. The engineer is strapped in place, facing upward. Just before the blade is released he calls out, "Wait a minute. I think I see your problem."

Engineers are problem solvers who want to see devices work correctly and work well. Most of us also need to know how things work.

We engineers are problem solvers. We want to see devices work correctly and work well. Most of us also seem to have a need to know *how* things work. As children, we took apart motors, radios, the family television set, and anything else we could lay our hands on. When we retire, we will tinker.

This month, we will set aside the theory and philosophy and take apart a fuzzy rule-based system to see, component by component, how one actually works.

For our purposes, "degree of membership" and "truth value" mean much the same thing. A 5-ft, 8-in. woman as a member of the set "tall women," which has a degree of membership of 0.80, is identical to the statement "A 5-ft, 8-in. woman is a tall woman" being true with a truth value 0.80. Keeping this in mind will help you wade through the following.

A fuzzy rule-based system includes three sections: the fuzzification step, or fuzzifier, which converts crisp input values into degrees of membership in fuzzy input sets; the inference step, or rule base, which determines from input conditions the actions to take; and the combination/defuzzification step, or defuzzifier, which combines multiple fuzzy actions and determines a single crisp executable output.

A fuzzy controller for a motor that drives the flow of a fluid provides an example to show the operation of each component. The controller has two sampled inputs, the motor's rotational velocity, ω, and the fluid's flow-rate error, ϵ, which is the difference between the actual and the commanded flow rates. The controller has one output, the motor-drive current, I_d. The example is incomplete and intentionally artificial because I want to concentrate on how the fuzzy system works and not on what it is doing.

Fuzzification

Let's first discuss the fuzzifier in terms of the input, ω, a crisp input with values ranging over $0\omega \leq \omega \leq 2000$ rpm. Because this is a fuzzy system, we will work with the fuzzy, or linguistic, values of the fuzzy input Rotational_Velocity. The system designer assigns these fuzzy values and gives them labels, such as Near_Zero, Very_Low, Low, Medium, High, and Very_High. Each of these labels represents a fuzzy set in the operational domain of possible crisp values (**Fig 1**). You can consider each set a (fuzzy) value of the (fuzzy) variable Rotational_ Velocity. You can define each set as a membership function, with domain (its "x-axis") over the possible crisp values and range (its "y-axis") from 0 to 1.

Actual measured values are crisp (for example, 1025 rpm). You express rules using fuzzy terms, such as "if Rotational_Velocity is High then..." You perform fuzzification, or transformation from crisp to fuzzy values, by identifying to what degree the crisp value is a member of each of the fuzzy sets.

PHOTOGRAPHY / LIANE ENKELIS

You typically use the label μ, ranging in value over $0 \leq \mu \leq 1$, to represent this calculated degree of membership. You calculate the degrees of membership of a crisp-input value directly from the (membership) functions that represent the fuzzy sets.

For example, ω=1025 rpm results in a degree of membership of 0.69 in the set High, or

$$\mu_{High}(\omega=1025)=0.69.$$

For successful operation of a fuzzy system, input membership functions overlap. The fuzzy set Very_High, therefore, also contains the same crisp value, ω=1025, or

$$\mu_{Very\_High}(\omega=1025)=0.31.$$

It has zero membership in all other sets (Fig 2). You now apply these degrees of membership as truth values to the conditions of rules in the inference step.

Rules

In rules, you express both conditions (antecedents) and actions (consequences) as fuzzy terms, most often in the form "if (condition) then (action)." An example is

IF Rotational_Velocity is Medium
AND
Flow_Rate_Error is Positive_Large
THEN Drive_Current is Med_Large;.

You calculate the degree to which each of the conditions is true as part of the fuzzification step and apply these as inputs to the rule base. If a rule contains several conditions linked by one or more logic operators, you use fuzzy-logic operators to combine the respective truth values. Fuzzy-logic operators, such as AND, OR, and NOT, are conceptually the same as bilevel logic operators, but you define fuzzy-logic operators differently. If you have two fuzzy truth values, μ_A and μ_B, you could define AND, OR, and NOT as:

$$\mu_A \text{ AND } \mu_B=\min(\mu_A, \mu_B)$$
$$\mu_A \text{ OR } \mu_B=\max(\mu_A, \mu_B),$$
$$\text{NOT } \mu_A=1-\mu_A$$

as Professor Zadeh originally defined them. Although others have proposed and used other fuzzy operators in various systems, the "Zadeh operators" are still the most popular. When μ_A and μ_B are either 0 or 1, these definitions are the same as the Boolean operators.

Any rule that has a nonzero value for its condition has "fired." Because of overlapping input membership functions, more than one rule often fires at a time. You do this to achieve smooth output transitions between regions where a single rule fires. Now comes a key to how a fuzzy rule base works. *The strength of the firing of each individual rule is the degree to which its conditions are true; you apply this strength as a truth value to the action or actions the rule indicates.* If a rule fires weakly, the action it specifies only weakly impacts the subsequent system output. If the rule fires strongly, that rule's action greatly affects the system output.

In the rule above, let "Rotational_ Velocity is Medium" have a truth value of 0.69; that is, the crisp rotational velocity input, ω, is a member of the set Medium with 0.69 degree of membership. Also, let "Flow_ Rate_ Error is Positive_Large" have a truth value of 0.35. The truth value of the rule's condition is the AND of the truth values of the two expressions and is, therefore, the minimum value, 0.35. Because this value is nonzero, the rule has fired, and the truth value of the condition is applied as the truth value of the action "Drive is Med_Large" receives the truth value 0.35. We use this value in the combination/defuzzification step.

Combination/defuzzification

Each rule that fires at each system iteration specifies its own action. A degree of membership in an output fuzzy set now represents each fuzzy action. Fig 3 shows what the output sets for the Drive_Current output in our example might be.

Two steps remain to determine the crisp, executable system output: combining all fuzzy actions into a single fuzzy action and transforming the resulting single fuzzy action into a crisp, executable system output. A single rule's firing requires no combination step, only defuzzification. Also, you can perform combination

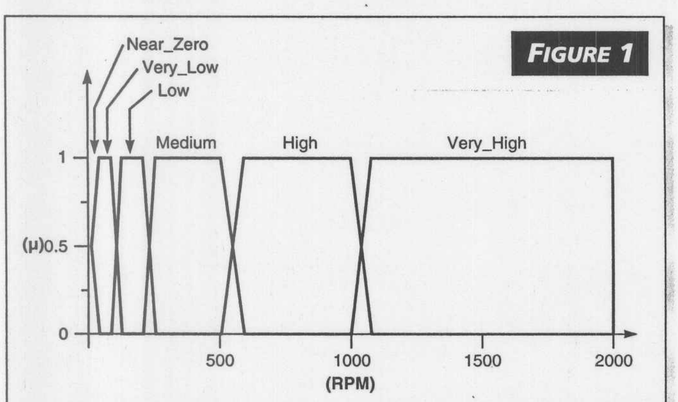


FIGURE 1

When you treat a variable as being fuzzy, its values are also fuzzy. Although the fuzzy system measures a crisp input value of 1025 rpm, it is more interested in whether and to what degree the measurement is High or Very_High.

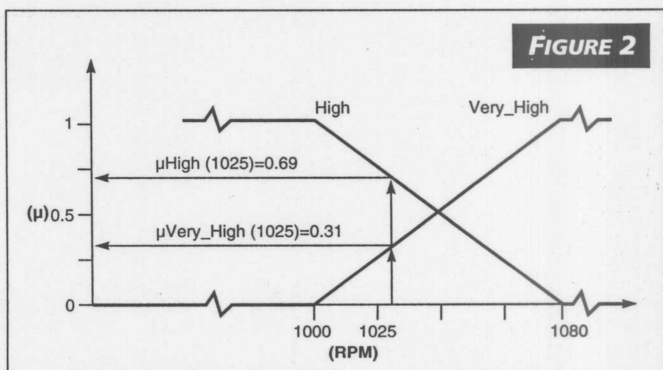
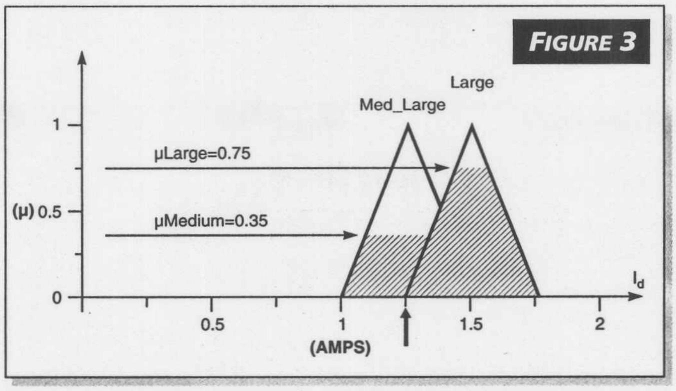


FIGURE 2

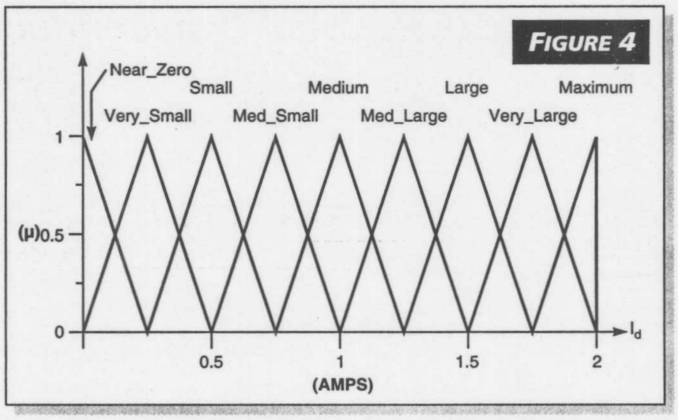
A measured value of 1025 rpm belongs to both High and Very_High sets. Solving the respective membership function determines the degree to which the value belongs to each set. Thus, for ω=1025, Rotational_Velocity is said to be High with degree of membership of 0.69 and Very_High with degree of membership of 0.31.

FIGURE 3



You can also define fuzzy actions linguistically and represent them as membership functions in the output space, in this case, over the range of possible drive currents.

FIGURE 4



If two triggered rules specify output actions of "Drive_Current is Med_Large" (μ=0.35) and "Drive_Current is Large" (μ=0.75), the centroid method of defuzzification lops the tops off the respective membership functions at the specified μ and then finds the center of mass of the centers of mass of the resulting areas. This is the same as treating the indicated regions as two metal plates, glued together and locating the balance point. In this example, the defuzzified output, which will execute, is approximately I_d=1.23A (heavy vertical arrow).

and defuzzification in either order or simultaneously.

The most popular combination/defuzzification technique is the "center-of-mass," or "centroid," technique in which each rule action specifies an output fuzzy value (such as "Drive_Current is Large") and a strength (the indicated truth value). In the centroid method, you top the membership function representing each fuzzy value at the value of the truth value, μ, with which the rule fired. You calculate the center of mass (centroid) of each resulting area and determine the center of mass of these centers of mass. The centroid of the combined areas defines the crisp, executable output (Fig 4).

You can most easily visualize this process by treating areas as rigid sheets that are glued together. The center of mass of the resulting collection of sheets, the point at which it will balance, identifies the crisp output value to execute.

The truth value of each output is not the strength with which it executes, but rather a weight in the combination process. In our example, if a single rule were to fire, the executed output would be the same, regardless of the strength with which it fired. This is because the centroid of a vertically symmetric output membership function is independent of the level at which it is lopped off.

To summarize, a fuzzy-rule-based system is a sampled data system that maps inputs to outputs through rules. The system samples inputs at each sampling interval and applies those inputs to the appropriate input membership sets to determine their degrees of membership in each set. This step is fuzzification.

You express how a system operates by its rules, typically using an "if-condition-then-action" format. The degrees of membership of inputs in fuzzy input sets are the rule conditions. You also express rule actions as fuzzy sets. The truth value of an action is the degree to which the corresponding input condition is true. You can logically combine multiple expressions into a condition using fuzzy-logic operators. Scanning the rule base for a set of inputs is the inference step.

When several rules fire at once, as is often the case in a fuzzy system, their actions must combine into a single action. In addition, because you express this single action as a combination of fuzzy sets, you must convert it back into a crisp executable output. This final step is combination/defuzzification, or, simply, defuzzification. This fuzzify-infer-defuzzify sequence executes at each system time increment for each new set of system inputs, each time generating a set of executable outputs. That is how a fuzzy-rule-based system works.

When I first told the guillotine joke to my family, my then-five-year-old daughter asked if the engineer also went free. I would like to think so, but probably not.

To close, I have two administrative announcements. First, starting next year I will intersperse the normal tutorials, application discussions, and tool reviews with an occasional question-and-answer exchange. You readers will be the source of questions. Please send them to me directly using either the postal address or electronic-mail address indicated below.

Second, I am attempting to respond to all who send in comments or questions and will continue to do so unless the numbers involved make it impossible. On several occasions, my response to an e-mailed message has been returned with the Internet equivalent of "addressee unknown." For those of you who correspond via e-mail, please include your postal mailing address; if my electronic response bounces, I will mail it to you. For those of you who have sent e-mail to me and have not received a response, please send it again and include your postal address.

David Brubaker is a consultant in fuzzy-system design. You can reach him at Huntington Advanced Technology, 883 Santa Cruz Ave, Suite 31, Menlo Park, CA 94025-4608 or on the Internet at: brubaker@cup.portal.com.